

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開平11-353205

(43) 公開日 平成11年(1999)12月24日

(51) Int.Cl.⁶

G 0 6 F 11/28

識別記号

3 1 0

F I

G 0 6 F 11/28

3 1 0 B

審査請求 未請求 請求項の数4 F D (全 7 頁)

(21) 出願番号 特願平10-170528

(22) 出願日 平成10年(1998) 6 月 4 日

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込 1 丁目 3 番 6 号

(72) 発明者 門脇幸男

東京都大田区中馬込 1 丁目 3 番 6 号 株式
会社リコー内

(72) 発明者 樽木杉高

東京都大田区中馬込 1 丁目 3 番 6 号 株式
会社リコー内

(72) 発明者 中村圭治

東京都大田区中馬込 1 丁目 3 番 6 号 株式
会社リコー内

(74) 代理人 弁理士 園田 敏雄

最終頁に続く

(54) 【発明の名称】 トレースメモリ内蔵のプロセッサ

(57) 【要約】

【課題】 L S I に搭載されている C P U や D S P 等のプログラム実行やデータアクセス等の動作履歴を格納する機能 (トレースメモリ) を同 L S I 内部に持たせる場合において、トレース用メモリの小容量化を実現し、高速な L S I に対してのリアルタイムトレースを実現できるように、トレース用メモリへ記憶させるべきトレース情報の圧縮方法を工夫すること。

【解決手段】 プログラムアドレスのトレースの際、シーケンシャルな動作または分岐動作の違いを特定ビットで表し、これをアドレスの代わりにトレースメモリに記憶し、分岐の際、パイプラインの段数だけ分岐先アドレスを分割し、パイプラインハザード期間にこの分割された分岐先アドレスと分岐を表す特定ビットをトレースメモリへ記憶させること。

【特許請求の範囲】

【請求項1】プログラムアドレスのトレースの際、アドレスのシーケンシャルな動作または分岐動作の違いを特定ビットで表し、これをアドレスの代わりにトレースメモリに記憶し、分岐の際、パイプラインの段数だけ分岐先アドレスを分割し、パイプラインハザード期間にこの分割された分岐先アドレスと分岐を表す特定ビットをトレースメモリへ記憶させることを特徴とするトレースメモリ内蔵プロセッサ。

【請求項2】プログラムアドレスのトレースの際、アドレスのシーケンシャルな変化の数をカウントするカウンタを有し、アドレスの分岐が発生した場合は分岐先アドレスと上記カウンタのカウント値をトレースメモリに記憶させることを特徴とするトレースメモリ内蔵プロセッサ。

【請求項3】プログラムアドレスのトレースの際、アドレスのシーケンシャルな変化に対してはその変化毎に“1”（または“0”）を1ビットづつシフトさせながらレジスタに書き込むとともに、アドレスの分岐が発生した場合、アドレスの先頭に“0”（または“1”）を付けた分岐先アドレスをシフトさせながら、さらにレジスタに書き込み、レジスタの全ての桁にデータが書き込まれたとき、上記レジスタのデータをトレースメモリに記憶させることを特徴とするトレースメモリ内蔵プロセッサ。

【請求項4】プログラムアドレスのトレースの際、シーケンシャルなアドレスの変化に対しては“1”（または“0”）を1ビットづつシリアルアクセスメモリで構成したトレースメモリにシフトさせながら書き込み、アドレスの分岐が発生した場合、アドレスの先頭に“0”（または“1”）を付けた分岐先アドレスを前のデータの後に引き続いてシフトさせながら上記シリアルアクセスメモリで構成したトレースメモリに書き込むことで、アドレス情報をアドレストレースメモリに記憶させることを特徴とするトレースメモリ内蔵プロセッサ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、LSIに搭載されているCPUやDSP等のプログラム実行やデータアクセス等の動作履歴を格納する機能（トレースメモリ）を同LSI内部に持たせる場合において、有効にトレースメモリを使用するための手段、方法に関するものであり、トレースメモリを最小限のメモリ容量で実現することができ、チップ面積を低減することができるものである。

【0002】

【従来の技術】一般的にソフトウェアの開発支援装置であるインサート・エミュレータ（以下ICE）の機能であるリアルタイムトレースは、ターゲットチップの外部にトレース用メモリを持たせることにより実現している。リアルタイムトレースは、ターゲットチップ内部

のCPUやDSPのアドレス、データ、ステータス等の情報（以下、トレースデータ）をトレースしたい期間だけ外部トレース用メモリに逐次書き込み保存する（当然ながらトレース用メモリの書き込み制御信号もターゲットチップから出力される）。トレース終了後そのデータを読み出し、パソコン等でソフト的に処理し、CRTに出力することによりターゲットチップの動作履歴を確認、解析することが可能となる。この場合、ターゲットチップの外部にトレース用メモリがあるため、トレースデータのビット数分の端子が、ターゲットチップに必要となる。例えば、アドレス32ビット、データ32ビット、ステータス8ビットであれば32ビット分の端子が別途必要となってしまう。また、アドレス、データが数系統ある様なCPU、DSPを使用している場合は非常に多くの端子が必要となってしまう、チップ面積は実際の回路規模ではなく端子数により決定されチップのコストが高くなってしまいう問題がある。また、ターゲットチップとトレース用メモリはICEボード上に実装されており、動作速度が非常に高速なチップではボード上の配線遅延、ノイズ、トレース用メモリの性能等によりトレース用メモリへの書き込みが間に合わず、正常に書き込むことができない場合があり、ターゲットチップとトレース用メモリとのインターフェース部を精度よく設計していても、実際にはボード実装後のデバッグ時間がかなり必要となっているという問題がある。近年ではこれらの問題から、ターゲットチップ内部にトレース用メモリを搭載することにより、ICE用端子の削減、ICEボードのデバッグ時間の短縮、高速動作時でのリアルタイムトレースが可能となっている。しかしながら、トレース用メモリをターゲットチップ内に搭載する場合において、例えば、実行プログラムのアドレスをトレース用メモリに格納する場合、実行プログラムアドレスのビット長、任意の実行プログラムのステップ数をワード長とするメモリが必要となり、トレース用メモリによるコスト増とリアルタイムトレース機能の確保とのトレードオフになる。そこで、分岐先アドレスと割り込み直前の実行アドレスのみをトレースメモリに格納する方法が特開平8-95945号公報に記載されている。この方法は、マイクロプロセッサにトレースメモリを持たせ、プログラムアドレスのトレースメモリへの格納は分岐命令、割り込み等に対して、実行終了命令のプログラムカウンタ値、現在実行している命令のプログラムカウンタ値を選択して格納するものであるが、分岐後のシーケンシャルな命令実行についてのトレース結果については格納されておらず、ICE側でのトレース結果データの整理、表示にあたり、分岐元アドレスを容易に特定することは困難である。また、分岐元アドレスと分岐先アドレスのみをトレースメモリに格納した場合の例を図6に示すが、この図のように1回の分岐でトレースメモリを2ワード分消費することになるという欠点がある。

【0003】

【本発明が解決しようとする課題】そこで本発明では、上記の問題を解決するために、トレース用メモリをターゲットチップ内部に持たせる場合において、トレース用メモリの小容量化を実現し、高速なLSIに対してのリアルタイムトレースを実現できるように、トレース用メモリへ記憶させるべきトレース情報の圧縮方法を工夫することをその課題とするものである。

【0004】

【課題解決のために講じた手段】

【解決手段1】トレースメモリ内蔵プロセッサについて、プログラムアドレスのトレースの際、アドレスのシーケンシャルな動作または分岐動作の違いを特定ビットで表し、これをアドレスの代わりにトレースメモリに記憶し、分岐の際、パイプラインの段数だけ分岐先アドレスを分割し、パイプラインハザード期間にこの分割された分岐先アドレスと分岐を表す特定ビットをトレースメモリへ記憶させること。

【作用】パイプラインの段数だけアドレスを分割し、パイプラインハザード期間に分割された分岐先アドレスをトレースメモリに格納する。また分岐命令かシーケンシャルな命令かの判別ビットを用いることで、分岐命令時には「0」を、シーケンシャルな命令を実行している時はプログラムアドレスを格納する代わりに「1」を格納する。以上のトレース結果から少ないメモリ容量で、多くの実行履歴を格納することが可能となる。またICE側にとっては、分岐先アドレス、判定ビットの「1」の数をカウントすることにより、分岐元アドレスを容易に特定することが可能となる。

【0005】

【解決手段2】トレースメモリ内蔵プロセッサについて、プログラムアドレスのトレースの際、アドレスのシーケンシャルな変化の数をカウントするカウンタを有し、アドレスの分岐が発生した場合は分岐先アドレスと上記カウンタのカウント値をトレースメモリに記憶させること。

【作用】シーケンシャルな命令をカウントする機能を持たせ、トレースメモリに対しては、分岐先アドレスとシーケンシャルな命令をカウントした値を格納する。以上のトレース結果から少ないメモリ容量で、多くの実行履歴を格納することが可能となる。またICE側にとっては、分岐先アドレス、カウント値により、分岐元アドレスを容易に特定することが可能となる。

【0006】

【解決手段3】トレースメモリ内蔵プロセッサについて、プログラムアドレスのトレースの際、アドレスのシーケンシャルな変化に対してはその変化毎に「1」（または「0」）を1ビットずつシフトさせながらレジスタに書き込むとともに、アドレスの分岐が発生した場合、アドレスの先頭に「0」（または「1」）を付けた分岐

先アドレスをシフトさせながら、さらにレジスタに書き込み、レジスタの全ての桁にデータが書き込まれたとき、上記レジスタのデータをトレースメモリに記憶させること。

【作用】シーケンシャルな命令についてはその判別ビットとして「1」を、分岐命令についてはその判別ビットとして「0」、および分岐先アドレスをトレースメモリに圧縮して格納する。以上のトレース結果から少ないメモリ容量で、多くの実行履歴を格納することが可能となる。またICE側にとっては、分岐先アドレス、判定ビットの「1」の数をカウントすることにより、分岐元アドレスを容易に特定することが可能となる。

【0007】

【解決手段4】トレースメモリ内蔵プロセッサについて、プログラムアドレスのトレースの際、シーケンシャルなアドレスの変化に対しては「1」（または「0」）を1ビットずつシリアルアクセスメモリで構成したトレースメモリにシフトさせながら書き込み、アドレスの分岐が発生した場合、アドレスの先頭に「0」（または「1」）を付けた分岐先アドレスを前のデータの後に引き続いてシフトさせながら上記シリアルアクセスメモリで構成したアドレストレースメモリに書き込むことで、アドレス情報をトレースメモリに記憶させること。

【作用】シーケンシャルな命令についてはその判別ビットとして「1」を、分岐命令についてはその判別ビットとして「0」、および分岐先アドレスをシリアルアクセスメモリで構成したトレースメモリに順次シフトさせながら圧縮して格納する。以上のようにシーケンシャルなトレースメモリを用いてトレースさせることにより、レジスタを用いることなく簡単な回路構成で高速に、しかも少ないメモリ容量で、多くの実行履歴を格納することが可能となる。またICE側にとっては、分岐先アドレス、判定ビットの「1」の数をカウントすることにより、分岐元アドレスを容易に特定することが可能となる。

【0008】

【実施例】本発明のトレースメモリ内蔵プロセッサを用いてデバッグを行うためのシステムの概念的な構成を図1に示している。デバッグ対象となるターゲットチップ100には、機器の制御やデジタル信号処理等の演算を行うためのCPU10とプログラム及びデータを格納するためのメモリ11を搭載している。従来、これらCPUとメモリ11は別々のチップで実現し、これらを基板上に置き接続することでシステムを構成していた。しかし近年半導体技術の進歩と共にチップの高速化と小面積化が実現でき、システムの小型化と低コスト化、さらに低消費電力化を図るために1チップにCPUとメモリ等を一緒に搭載するケースが増えてきている。このようなシステムでプログラムやデータのデバッグを行うためには、CPU10から出力されるメモリ11へのアドレ

ス20とCPU10とメモリ11間でやりとりされるデータ21をリアルタイムに監視し、プログラムやデータの誤りを見つけ、修正することが必要である。このため一般的には、ターゲットチップ外部のデバッガ16、すなわちICE（インサーキット・エミュレータ）にターゲットチップ100内のアドレス20とデータ21を取り出し、プログラムやデータの検証をとる方法が用いられる。しかし、高速に動作するシステムでは、ターゲットチップ10からアドレス20やデータ21の信号を取り出す際に波形がなまり、正確な信号が取り出せないといった問題やアドレスやデータ幅が大きい場合、例えば32ビットプロセッサではアドレス32ビット、データ32ビットでこれら両方の信号を取り出すには、ターゲットチップ100に計64ビットの端子が新たに必要になり、高価な多ピンパッケージを使用せざるを得ないといった問題があった。またリアルタイムデバッグにおける波形のなまりの問題を解決するため、ターゲットチップ100内にトレースメモリを内蔵し、リアルタイムで動作させる際、一旦アドレス20やデータ21をトレースメモリに格納し、後で比較的低速な動作で外部のデバッガにアドレスやデータを取り出すということも行われている。しかし、この場合でもある時間内のアドレスとデータを格納するには、容量の大きいトレースメモリが必要であるといった問題があった。デバッグのために容量の大きいトレースメモリを搭載してもデバッグが終わったシステムでは、トレースメモリは不要であり、これがあるためにチップ単価が高くなったり、またトレースメモリを搭載したチップと搭載しないチップの2種類を作成し、デバッグを行うときはトレースメモリを搭載したチップを使い、デバッグが終了し、実際の製品に組み込む場合はトレースメモリを搭載しないチップを使用するという方法もあるが、このためには2種類のチップを作成することになるためにコストがかかるという問題もある。

【0009】本発明は、以上の問題を極めて効果的に解消したものであり、トレースメモリの容量をできるだけ少なくしてデバッグ可能なターゲットチップを提供することができるものである。本発明のトレースメモリ内蔵プロセッサを用いてデバッグを行うためのシステムの概念的な構成を示す図1において、特徴的な点はターゲットチップ100内のアドレス用のトレースメモリ14の入力側に符号化回路12を設けており、またデバッグ16の入力側に復号化回路15を設けていることである。これによりトレースメモリ14にアドレスを格納する際は、CPU10からメモリ11へ出力されるアドレス20の変化を利用し、できるだけメモリ容量を削減できるようにアドレスの符号化を行い、この符号化データをトレースメモリに格納する。また、デバッグ16に取り込む際には、この符号化データを元に戻し（復号化）て検証を行う。次にこの方法を具体的に示す。図2は、メモ

リアドレス20の変換方式の一例を示したものである。t0～t13は、サイクル0からサイクル13を示す。ビットBは、分岐状態であるか否かを示し、‘1’のとき分岐なし、‘0’のとき分岐ありを示す。一般的に分岐ありの場合は、アドレスが‘1’ずつ増加されず、今までと異なったアドレス値を示すことでこれを認識できる。従って、アドレス値が順に‘1’ずつ増加している場合は、分岐は行われず、命令を順に実行していることを示す。D7～D0は、分岐先アドレスを格納するフィールドで、図2では1サイクルに8ビットを格納できる場合を示している。例えば、サイクルt5でBビットが‘0’となり分岐したことを示しているが、この分岐先アドレスは、サイクルt5～t8のD7～D0ビットで表される。この例ではアドレスは32ビットで、サイクルt5における分岐先アドレスは2進数で、‘01101011, 11000111, 01101110, 00001100’であることを示している。ちなみにサイクルt5～t8で分岐先アドレスを格納する場合は、最も上位（左側）のビットから格納している。

【0010】以上を図5に示すタイミングチャートで説明する。図5はプロセッサの実行過程が、命令フェッチ、デコード1（プリデコード）、デコード2、実行の4段パイプライン動作の場合を示している。ここでアドレス100はアドレス200への分岐命令であり、図2のサイクルt5に対応している。このアドレス100の分岐命令は実行ステップで分岐が決定するとすぐにアドレス200の命令からフェッチを開始する。そしてその間フェッチされたアドレス101, 102, 103の命令は実行すべきでない命令であり、不要となる（パイプラインハザード）。これは図2におけるサイクルt6～t8に対応する。このように分岐命令とその後のパイプラインハザードのサイクルを利用して、従来、例えば32ビットで表される分岐先アドレスを8ビットずつに分割して、これを4サイクルかけてメモリに格納することでメモリを効率良く使用することができる。図3は、別の符号化方式を示したものである。これは、アドレスを格納するフィールドを32ビット分用意し、分岐命令間の一般命令の数も格納することで分岐命令の分岐先アドレス及び分岐命令以外の通常命令がどれだけ実行したかを格納できる。例えば図3の例では、通常命令を3命令実行した後に分岐命令を実行しており、その分岐先アドレスは、分岐アドレス1で示される。その後、通常命令が6命令続いた後に、分岐命令が実行され、その分岐先アドレスは分岐アドレス2で示される。このような手順によりアドレスの変化をトレースメモリに蓄積することでトレースメモリの容量を削減することができる。

【0011】図4は、別の実施例を示したもので、シリアルなビット列としてアドレス情報をトレースメモリに格納する場合を示している。アドレス情報格納の開始は、トレースメモリの左上となっており、ビット方向に

格納していき、ビット方向が全て埋まったらワード方向に1行下がり、左から右へ格納するという手順を踏む。シフトはビット方向に格納する際の位置合わせをするもので、レジスタは一時的にデータを保持するためのもので、レジスタにビットが埋まったときにメモリに1行分を書き込む。メモリ書き込み後はレジスタの値はリセットされ、次のアドレス情報が格納される。図4の例では、スタートビットから右側に通常命令を示すビット‘1’が5ビットあり、6ビット目に分岐命令を示すビット‘0’がある。このことは通常命令を5命令実行した後に、分岐命令を実行しており、分岐命令の分岐先アドレスは分岐命令を示すビット‘0’の後の決まった長さで示される。図4の場合、分岐先アドレスが10ビットの場合を示している。また、上記図4の実施例において、トレースメモリとして、シリアルアクセスメモリで構成したトレースメモリを用い、シーケンシャルな命令についてはその判別ビットとして‘1’を、分岐命令についてはその判別ビットとして‘0’、および分岐先アドレスを上記シリアルアクセスメモリで構成したトレースメモリに順次シフトさせながら格納するようにすれば、レジスタを用いることなく簡単な回路構成で高速に、しかも少ないメモリ容量で、多くの実行履歴を格納することが可能となる。以上により、トレースメモリを有効かつ最小限のメモリ容量で実現でき、プログラムアドレスの推移の情報を格納することができる。

【0012】

【効果】(1) 解決手段1による発明(請求項1に係る発明)について

プログラムアドレスのトレースの際、アドレスがシーケンシャルな動作をしているか、それ以外の動作をしているかを特定ビットで表すことにより、すべてのアドレスをトレースするときよりもメモリ容量を削減することが可能となり、さらに、分岐先のアドレスをパイプラインハザード期間に分割してトレースメモリに記憶させることにより、分岐先のアドレスの全ビットを一括して記憶させる場合よりも、メモリ容量を削減することも可能となり、チップ面積を縮小することができる。

(2) 解決手段2による発明(請求項2に係る発明)について

プログラムアドレスのトレースの際、アドレスがシーケンシャルな動作をしている場合をカウントしたカウンタ

値と分岐先アドレスのみをアドレ스트レースメモリに記憶させることにより、アドレスのメモリ容量を削減することが可能となり、チップ面積を縮小することができる。

(3) 解決手段3による発明(請求項3に係る発明)について

プログラムアドレスのトレースの際、シーケンシャルな動作をしているときのアドレス変化に対する情報“1

(または0)”と、分岐が発生した場合、アドレスの先頭に“0(または1)”を付けた分岐先アドレスのみをアドレ스트レースメモリに記憶させることにより、アドレスのメモリ容量を削減することが可能となり、チップ面積を縮小することができる。

(4) 解決手段4による発明(請求項4に係る発明)について

プログラムアドレスのトレースの際、シーケンシャルな動作をしているときのアドレス変化に対する情報“1

(または0)”と、分岐が発生した場合、アドレスの先頭に“0(または1)”を付けた分岐先アドレスのみを

シリアルアクセスメモリで構成したアドレ스트レースメモリに順次記憶させることにより、回路構成を簡素化でき、高速化が可能で、かつアドレスのメモリ容量を削減することが可能となり、チップ面積を縮小することができる。

【図面の簡単な説明】

【図1】は、本発明のトレースメモリ内蔵プロセッサを用いてデバッグを行うためのシステムの構成図。

【図2】は、パイプラインハザード期間を利用してアドレス情報をトレースメモリに格納する際のメモリの内容を示す図。

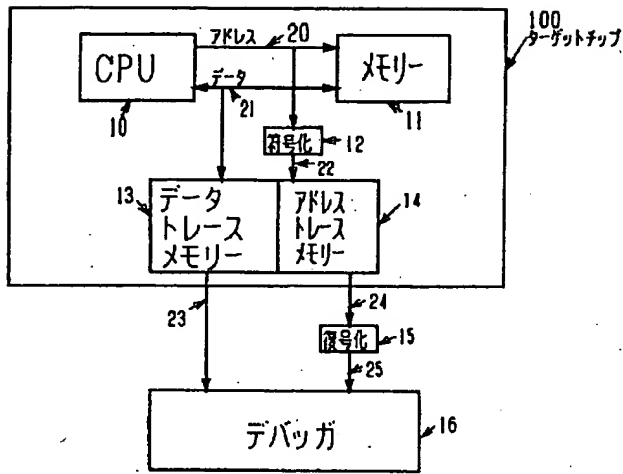
【図3】は、分岐アドレスと連続アドレスのカウント値をトレースメモリに格納する際のメモリの内容を示す図。

【図4】は、インクリメント情報と分岐アドレスをシリアルにトレースメモリに格納するための構成及びメモリの内容を示す図。

【図5】は、4段のパイプライン動作を示すタイミングチャート。

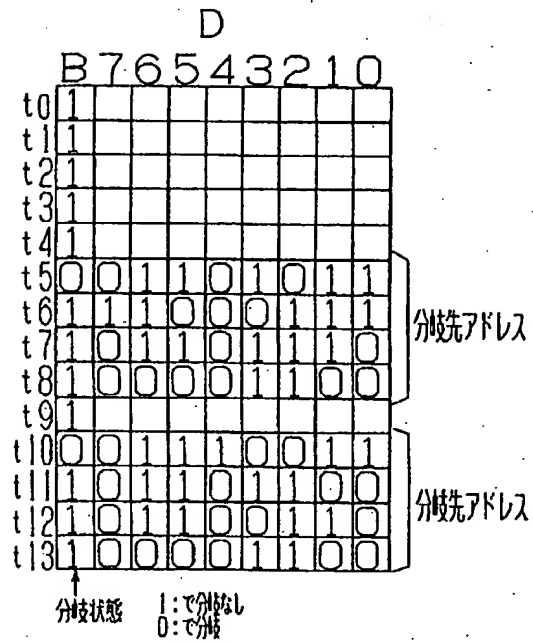
【図6】は、分岐元アドレスと分岐先アドレスのみをトレースする従来のトレースメモリの内容を示す図。

【図 1】



本発明の構成図

【図 2】



パイプラインハザードを利用してアドレス情報を保存

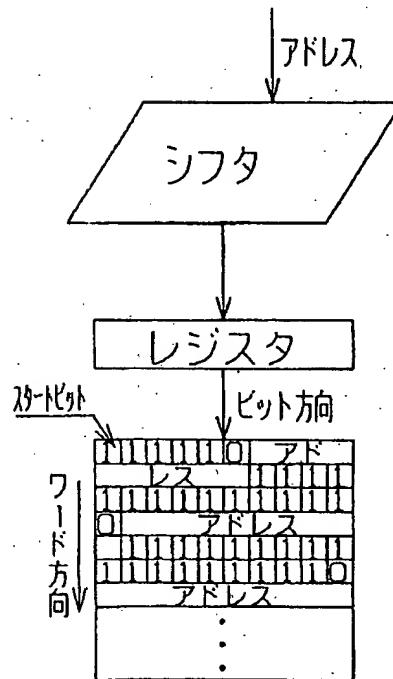
【図 3】

3	分岐アドレス1
6	分岐アドレス2
4	分岐アドレス3
2	分岐アドレス4
	分岐アドレス5
	分岐アドレス6
	分岐アドレス7

前の分岐アドレスから
今の分岐アドレスまでの
インストラクション数 (ランレングス)

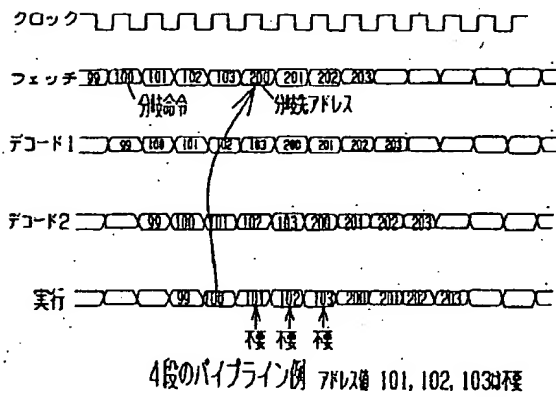
分岐アドレスと連続アドレスのカウンタ値を保存

【図 4】



インクリメント値と分岐アドレスをシリアルに保存

【図 5】



【図 6】

トレースメモリー アドレス	分岐先アドレス	
	分岐元アドレス	分岐先アドレス
0	00000100	00000200
1	00000250	00000430
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		

従来例 分岐先アドレスと分岐先アドレスのみトレース

フロントページの続き

(72) 発明者 平井敬康

東京都大田区中馬込 1 丁目 3 番 6 号 株式
会社リコー内